



File I/O

CSE 160

Summer 2026

Questions?



Announcements

- [Homework Assignment 2](#) due tonight at 11:59 PM
- [Programming Practice 3](#) due Sunday, July 19th at 11:59pm
- [Homework Assignment 3](#) due Monday, July 20th at 11:59pm
- [Practice Midterms](#) now available on the course website
 - Midterm on Monday, July 20th during class (9:40AM-10:40AM)
- Grades for Homework 1 will be released later today
 - We will make an announcement with **Resubmission 1** instructions!

Midterm Exam

Logistics

- Held in-class on Monday, July 20th (in HRC 155 - this room!)
- You **may** bring **one sheet of 8.5 x 11 inch paper as your own cheat sheet**
- You will be provided with an additional reference sheet as part of the exam packet

Resources

- **Past midterm exams** (+ answer keys) on [course website midterm page](#)
- Office Hours + Ed :)

Questions?



Today's Roadmap

1. What's in a File?
2. File Systems
3. Reading Files
4. Writing to Files

[Jupyter Hub](#)

What's a File?

A **file** is a **collection of data** stored in a **structured format**.

According to all known laws of aviation, there is no way a bee should be able to fly. Its wings are too small to get its fat little body off the ground. The bee, of course, flies anyway because bees don't care what humans think is impossible. Yellow, black. Yellow, black. Yellow, black. Yellow, black. Ooh, black and yellow! Let's shake it up a little. Barry! Breakfast is ready! Coming! Hang on a second.

text.txt

Full Name: _____
 Email Address (UW Net ID): _____ @u.w.edu
 Section: _____

CSE 160 Winter 2025 - Midterm Exam

Instructions:

- You have the **entire class period** to complete this exam.
- The exam is closed book, including no calculators, computers, phones, watches or other electronics.
- You are allowed a single sheet of notes for yourself.
- We also provide a syntax reference sheet.
- Turn in all sheets of this exam, together and in the same order when you are finished.
- When time has been called, you must put down your pencil and stop writing. Points will be deducted if you are still writing after time has been called.
- You may only use parts and features of Python that have been covered in class up to this point.
- You may ask questions by raising your hand, and a TA will come over to you.

Good luck!

Question	Topic	Points
Question 1	Expressions	20
Question 2	File I/O	10
Question 3	Nested Lists	8
Question 4	Dictionaries	8
Question 5	Functions	14

exam.pdf

41, 45, 50,
 46, 43, 43, 39, 41, 48, 48,
 46, 52, 54, 48, 46, 46, 46,
 45, 45, 45, 43, 39, 41, 39,
 41, 45, 46, 48,

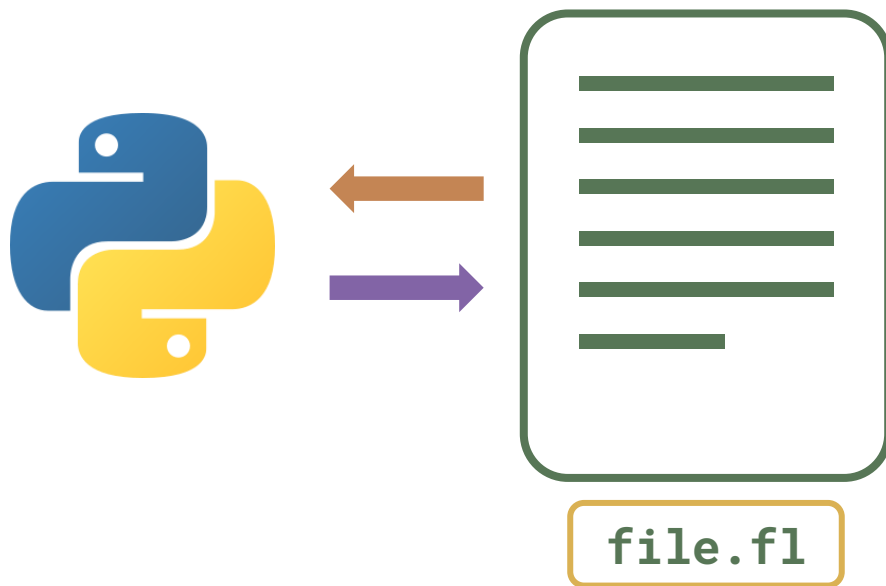
data.dat

Using Files

A **file** is a **collection of data** stored in a **structured format**.

To use a file, we might want to:

- **Input**: get data *from* the file
- **Output**: store data *to* the file



In order to do this, we need to know the file's **name** and its **location**

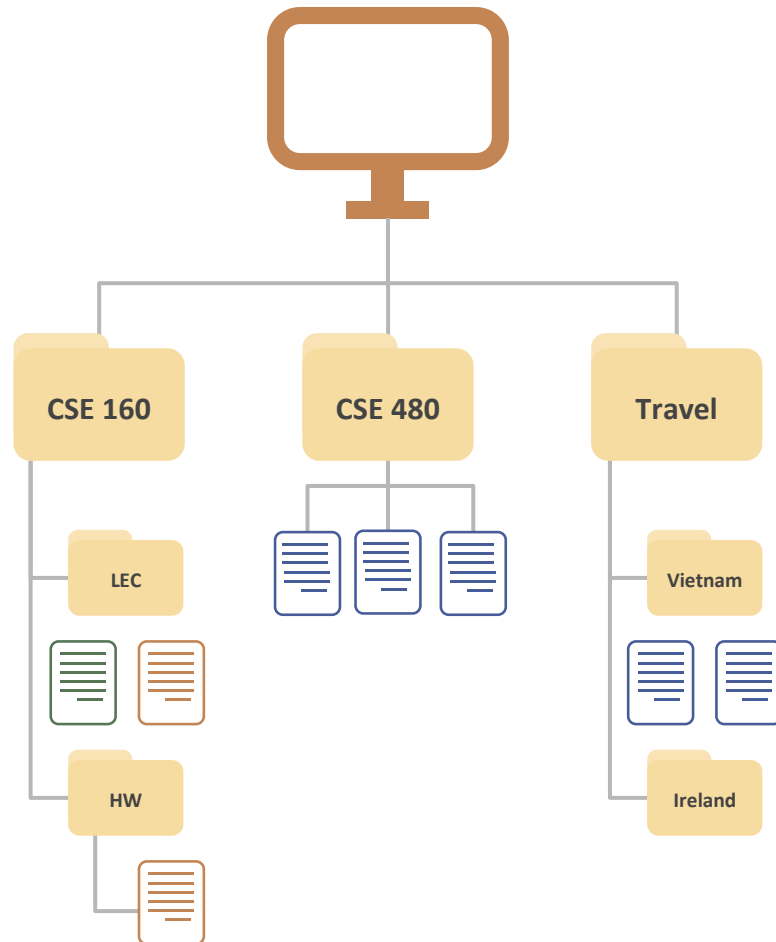
File Systems

A file system is a **hierarchical structure** that organizes files using (nested) **folders**

A file **directory** is a location (**folder**) within the file system:

```
C:\Users\adrian\Desktop\$
```

```
asalguer:~/COURSE_MATERIALS/lectures/lec09/$
```



Navigating a File System

A file system is a **hierarchical structure** that organizes files using (nested) **folders**

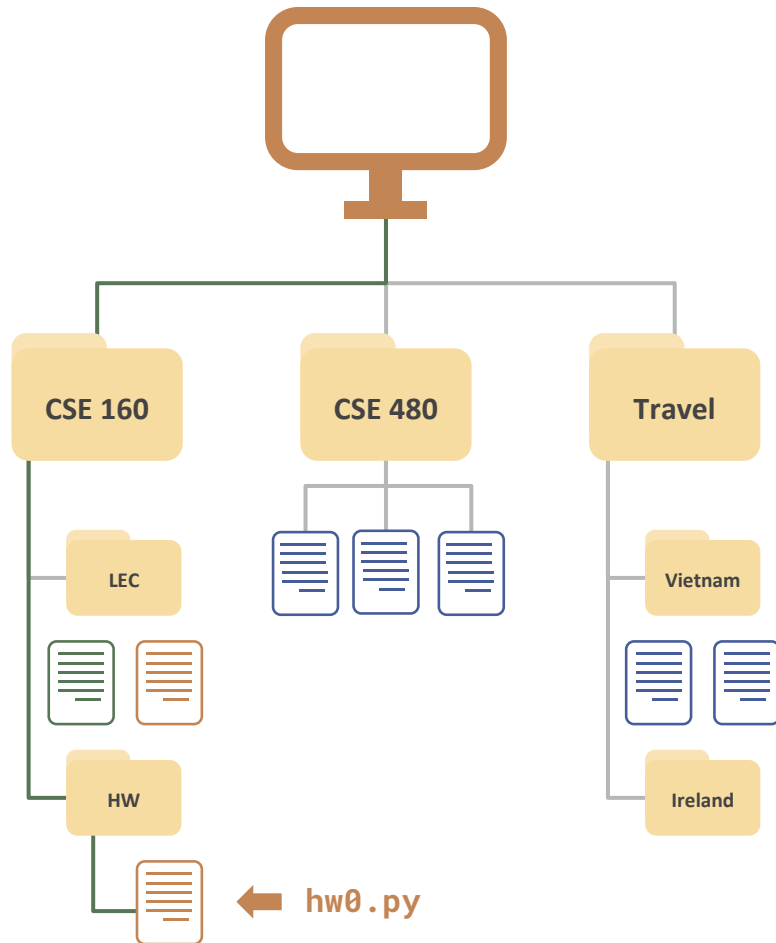
A file **directory** is a location (**folder**) within the file system

To locate a file within a directory, we need to specify its **path** from the directory:

```
... ~/Desktop/$ python ./'CSE 160'/HW/hw0.py
```

current directory

file path



Using the Terminal

A file **directory** is a location (**folder**) within the file system

We can use the **terminal** to navigate file systems:

cd *change **directory** to **path***

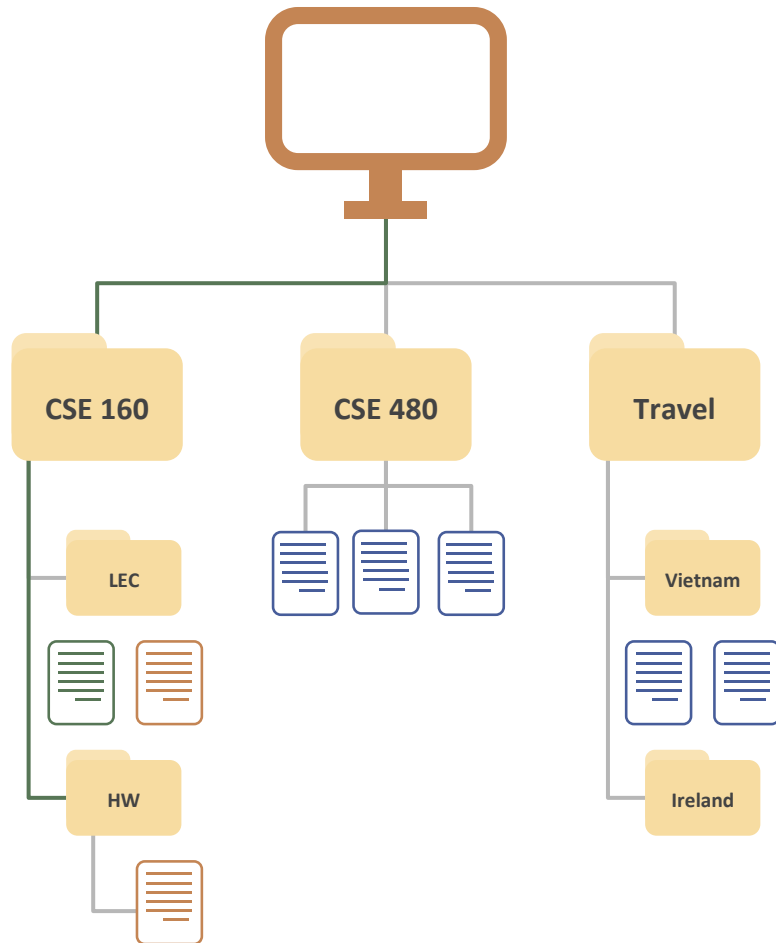
ls *list contents of current **directory***

```
... ~/Desktop/$ cd ../'CSE 160'/HW/
```

current directory

path to new
directory

CSE 160: File I/O



Questions?

Live and on Sli.do

Sli.do

#cse160



Think Pair Share

What command should Adrian type to run the **lec09.py** Python file from his current directory?

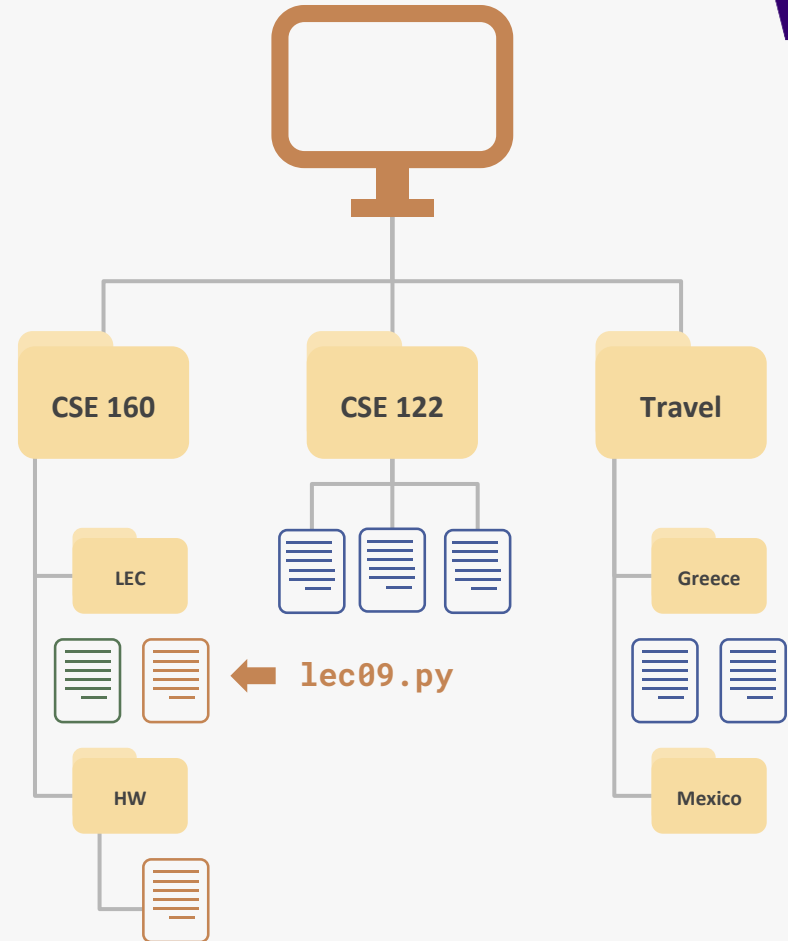
```
... ~/Desktop/CSE 160/$
```



Sli.do

#cse160

CSE 160: File I/O



Using Files in Python

In Python, to **load** a file into memory, we need to call the **open()** function.

```
file = open("file.fl")
```

*file path from directory
with Python file*



file.fl

The **open()** function returns a **file object**.

```
type(open("file.fl"))
```



<class '_io.TextIOWrapper'>

Reading Files

In Python, to **load** a file into memory, we need to call the **open()** function.

```
file = open("file.fl")
```

*file path from directory
with Python file*

The **open()** function **returns** a **file object**.

```
type(open("file.fl"))
```



```
<class '_io.TextIOWrapper'>
```



```
file.read()
```

The **read()** method **returns** the entire file contents as a **string**.

Reading Files (Option 2)

In Python, to **load** a file into memory, we need to call the **open()** function.

```
file = open("file.fl")
```

*file path from directory
with Python file*

The **open()** function returns a **file object**.

```
type(open("file.fl"))
```



```
<class '_io.TextIOWrapper'>
```

To **read** the contents of a file, one line at a time:



```
for line in file:  
    print(line)
```

The **file object** can be **iterated** over!

Writing to Files

To **load** a file into memory and **write** to it, call the **open()** function with **"w"**:

```
file = open("file.fl", "w")
```

*mode argument specifies how the file will be used.
default = "r" (read)*

Caution: when in **write mode**, the existing file contents are **overwritten**!

To **write** content to a file:



```
file.write(string)
```

The **write()** method writes to the file with the **string string**.

Appending to Files

To **load** a file into memory and **append** to it, call the **open()** function with "a":

```
file = open("file.fl", "a")
```

When in **append mode**, the existing file contents are **not overwritten**, just added to.

To **append** content to a file:



```
file.write(string)
```

The **write()** method writes to the file with the **string string**.

Cleaning Up After Yourself

When you're finished with a file:

```
file.close()
```

WHY?

1. Changes to a file may not be fully implemented until `close()` is called.
2. Keeping a file open may prevent other applications from using the file

Questions?

Live and on Sli.do

Sli.do

#cse160



Files Example 1: Eager Editor

Before

Hello!
I am a file :)

file.fl

What will happen?

```
f = open("file.fl", "r")  
  
print(f.read())  
f.write("New content :)")  
  
f.close()
```

ERROR! Cannot write to read-only file

After

Hello!
I am a file :)

file.fl

Files Example 2: Do Over

Before

Hello!
I am a file :)

`file.fl`

What will happen?

```
f = open("file.fl", "w")  
  
f.write("New!")  
f.write("Scratch.")  
  
f.close()
```

After

New!Scratch.

`file.fl`

Newline Characters

```
f.write("New content!")  
f.write("Scratch that.")
```



New Content!Scratch that.

When writing to a file, we have to **specify every character** to be written.

`\n`

"`\n`" is a **newline character**, that creates a new line.

```
f.write("New content!\n")  
f.write("Scratch that.")
```



New Content!
Scratch that.

Files Example 3: So Good I Read It Twice

Before

Hello!
I am a file :)

file.fl

What will happen?

```
f = open("file.fl", "r")  
  
print(f.read())  
print(f.read())  
  
f.close()
```

Reading a file **'consumes'** the contents of the file object.

Output

Hello!
I am a file :)

Think Pair Share

Write a program that reads in the contents of the file **nominees.txt** and **creates a list** containing each movie title as strings.



Sli.do

#cse160

Frankenstein
Sinners
F1
Hamnet
Bugonia
Sentimental Value
Train Dreams
The Secret Agent
Marty Supreme
One Battle After
Another

nominees.txt

Processing Data from Files

Frankenstein
Sinners
F1
Hamnet
...

nominees.txt

What now?

41, 45, 50,
46, 43, 43, 39, 41, 48, 48,
46, 52, 54, 48, 46, 46, 46,
45, 45, 45, 43, 39, 41, 39,
41, 45, 46, 48,

temps.dat

```
lst = [
    'Frankenstein\n',
    'Sinners\n',
    'F1\n',
    'Hamnet\n',
    'Bugonia\n',
    'Sentimental Value\n',
    'Train Dreams\n',
    'The Secret Agent\n',
    'Marty Supreme\n',
    'One Battle After Another'
]
```

```
lst = [
    '41, 45, 50,\n',
    '46, 43, 43, 39, 41, 48, 48,\n',
    '46, 52, 54, 48, 46, 46, 46,\n',
    '45, 45, 45, 43, 39, 41, 39,\n',
    '41, 45, 46, 48,'
]
```